Ontology Driven Sensor Independence in a Query Supported C_2-System

Erland Jungert
FOI (Swedish Defence Research Agency)
Box 1165
S-581 11 Linköping
SWEDEN

jungert@foi.se

Karin Silvervarg
FOI (Swedish Defence Research Agency)
Box 1165
S-581 11 Linköping
SWEDEN

karin@foi.se

Tobias Horney
FOI (Swedish Defence Research Agency)
Box 1165
S-581 11 Linköping
SWEDEN

tobho@foi.se

ABSTRACT

Information systems requiring sensor data input must generally include means for sensor data fusion as well as powerful mechanisms for user interaction and result visualization. From a user perspective the use of sensor data requires knowledge about the attached sensors and the data they generate. However, most end-user do not possess this kind of knowledge and for this reason techniques for sensor data independence must be developed to ensure that the users easily can apply their queries and interpret the result without too much trouble. In this work, an information system including a query language that can be applied to multiple sensor data is described. The query language allows data fusion when ever necessary. To facilitate query composition the query language has been furnished with a visual user interface that allows the end-users to apply their queries in a simple way. Furthermore, the query language has also been supplied with means for sensor data independence to make it possible for the end-users to apply queries without specific sensor and sensor data domain knowledge. The sensor data independence is made possible by an ontological knowledge base system.

1.0 INTRODUCTION

In current sensor-based information systems detailed knowledge about the sensors is required. Therefore sensor selection has been left to the users who supposedly are sensor experts. However, in real life this is not always the case. A user cannot be an expert on all sensors and sensor data types. Therefore systems with the ability to hide this kind of low-level information from the users need to be developed. Powerful user interfaces also need to be designed to allow the users to formulate queries with ease and request information at a high level of abstraction to accomplish sensor data independence. In the query language designed for multiple sensor data sources described in this work an approach to overcome these problems and to accomplish sensor data independence is proposed through the introduction of some novel concepts; among these are *the sensor dependency tree*, *the query refinement technique, the multi-level view data-bases*, and *an ontological knowledge base* for determination of the sensor algorithms. The concept of sensor data independence as introduced in this work should be seen from a user perspective where the purpose is to allow the end-users to apply queries concerned with an environment registered by multiple sensors by using well-known notions such as *area of interest*, *time interval of interest* and *requested object type*. Clearly, it is simpler to design user interfaces based on this type of notions instead of more complex types that must deal with the sensor data from specified sensors, which require a higher degree of domain knowledge. In sensor-based information systems [1][2] that include facilities for data fusion no concept similar to sensor data independence has yet been suggested. There are many reasons for this, for instance, users are expected to be sensor domain experts. This is, however, not always the case and since the users

Paper presented at the RTO IST Workshop on "Massive Military Data Fusion and Visualisation: Users Talk with Developers", held in Halden, Norway, 10-13 September 2002, and published in RTO-MP-105.

Report Documentation Page			Form Approved OMB No. 0704-0188
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.			
1. REPORT DATE **00 APR 2004**	2. REPORT TYPE **N/A**		3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Ontology Driven Sensor Independence in a Query Supported C2-System**			5a. CONTRACT NUMBER
			5b. GRANT NUMBER
			5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)			5d. PROJECT NUMBER
			5e. TASK NUMBER
			5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **FOI (Swedish Defence Research Agency) Box 1165 S-581 11 Linköping SWEDEN**			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release, distribution unlimited**			
13. SUPPLEMENTARY NOTES **See also ADM001665, RTO-MP-105 Massive Military Data Fusion and Visualization: Users Talk with Developers., The original document contains color images.**			
14. ABSTRACT			
15. SUBJECT TERMS			

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**29**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

are still not sufficiently competent they have not yet been able to specify and request this type of system property. Furthermore, this area is still somewhat immature with respect to the design and development of information systems with sensor databases. An important observation that needs to be pointed out is that while we in this work discuss the concept of sensor independence this also involves sensor data independence. Subsequently, we therefore talk about sensor data independence but this involves for the most part sensor independence as well although the latter concept may not be explicitly mentioned. This is motivated by the fact that these two concepts must go together.

The query language, ΣQL [3][4] can be seen as a tool for handling spatial/temporal information including means for sensor data fusion on data from multiple sensors. A query language of this type will use too complex query structures unless means to ensure simplification of the way the queries are defined can be made available. The query language is based upon a single operator type, i.e. the σ–operator that leads to a query structure with a relatively high degree of simplicity. Another, somewhat less important advantage of the concept is the natural and simple mapping of ΣQL-structures into an SQL-like query language. However, the SQL-like approach is primarily useful just in theoretical investigations, while the σ–query language that is easy to implement also is preferable because it is a step towards a user-friendly visual query language.

The work described in this paper is organized as follows. In section two the underlying concept of ontology driven sensor data independence is discussed. The query language and its general properties are more deeply discussed in section 3. The ontological knowledge base system that plays a fundamental role in the query system is described in section 4 while the user interface is discussed in section 5. Eventually the conclusions of the work are drawn in section 6.

2.0 ONTOLOGY DRIVEN SENSOR DATA INDEPENDENCE

Sensor data independence relates basically to data independence as introduced in database design where data independence was first introduced to allow modifications of the physical databases without affecting the application programs [5]. This was a very powerful innovation in information technology. The main purpose was to simplify the use of the databases from an end-user's perspective while at the same time allow a more flexible administration of the databases themselves [6]. A sensor based information system with properties of sensor data independence similar to the data independence in traditional databases would for similar reasons be an advantage.

A more serious motivation for sensor data independence depends on the extremely large data quantities that will be generated by the multi-sensor data systems. These data sources, i.e. the sensors, generally create heterogeneous data in large quantities. Thus it will in the future become more or less impossible to visualize these data individually with respect to their type, in such a way that the users will be able to extract all relevant information by means of the queries. That is, the users will not be able to identify objects of interest and even less so when multiple sources are available and fusion is the only way to determine a reliable result. This situation becomes even more obvious when considering that in many cases the users need responses to their queries quickly and in a suitable way because otherwise the workload of the users will be extremely high. Despite this, in many cases inexperienced users request raw data from single or multiple sensors without reflecting over whether this is realistic or whether they will be able to analyze these large data volumes in the first place. For these reasons sensor data independence is a necessary property of most future sensor data information systems. A very obvious consequence of the sensor data independence concept is that the user must learn to trust the system and the result presented to the users. Another important advantage with the sensor independence is that new sensor types can be integrated by just updating the ontological knowledge-base and if required include new recognition algorithms into the system. Thus integration of new sensor types can be done without informing the end-user.

Of main concern is how to design a system with the property of sensor data independence, which is one of the main research problems in this work. Similar to the data independence that is at hand in traditional databases sensor data independence must include a conceptual object description similar to a database schema. The conceptual object description must support the choice of sensors and sensor data algorithms and it must also be possible to tell which information that can be registered by a certain sensor. Such a structure can be described in terms of an ontology, e.g. [7]. However, the ontology is by no means sufficient by itself. To carry out the sensor and algorithm selections a mechanism that by means of the *requested objects, the actual weather conditions, the time of the day*, *the available sensor data* and the ontology must be available. For this a rule-based approach has been designed which will be discussed further in section 4.

3.0 ΣQL

The query language, ΣQL can be seen as a tool for the handling spatial/temporal information for sensor-based information fusion, because most sensors are generating spatial information in a temporal sequential manner. A query language of this type must be able to handle large data volumes because most sensors generate large quantities of data in very short periods of time. Another aspect to consider is that user queries may be concerned with data from more than one sensor, which consequently will lead to complex query structures, because the use of data from more than one sensor may require methods for multiple sensor data fusion.

3.1 The Query Language Structure

The strength of the query structure is its simplicity: the query language is based upon a single operator type, i.e. the σ–operator. The σ–query language can be seen as tool applied to the data sources and corresponding to a multidimensional space. This *source, R*, is also called a *universe.* Each query is made up by a sequence of σ-operators that primarily should allow operations on a sensor-data-independent level, i.e. the acquired sensor data should be transformed into a unified information structure at a high abstraction level that is sensor independent. To accomplish this, the queries should be expressed in terms of operator sequences where the operators carry out the transformations stepwise. Basically, the operators reduce the dimensions of the multi-dimensional search space to which each new operator is applied with respect to the dimensions in focus of the query. The reduced search space is subsequently called a *cluster*. Thus, as new operators are applied, the clusters become more and more refined until eventually a final cluster is returned and this cluster corresponds to the answer of the applied query.

An illustration of the query language could, for instance, be a video sequence, i.e. the universe R, from which a limited set of frames can be extracted. Thus if we are interested in three frames at different predetermined times, t_1, t_2, and t_3, along the time axis, this will correspond to the σ-operator σ_t (t_1, t_2, t_3), which means that the three frames should be selected from the time axis of the universe R. However, the main purpose of the operator sequences is to serve as an intermediate representation between the graphical user interface and the query interpreter. Evidently, this cannot be seen as a reasonable query technique for the end-users and thus a more user-adapted approach is needed. This will be discussed subsequently in section 5.

3.2 The Sensor Dependency Tree

In database theory, query optimization is often performed with respect to a query plan where the nodes represent the various database operations to be performed [8]. The query plan can be transformed in various ways to optimize query processing with respect to certain cost functions. In sensor-based query processing, a concept similar to the query plan is proposed. It is called the *sensor dependency tree*, which is a tree in which each node P_i has the following parameters:

object   is the object type to be recognized

source   is the information source

$recog_i$ is the object recognition algorithm to be applied

aoi_i is the spatial area-of-interest for object recognition

ioi_i is the temporal interval of interest for object recognition

$time_i$ is the estimated computation time in some unit such as seconds

$range_i$ is the confidence range in applying the recognition algorithm represented by two numbers min and max from the closed interval [0,1]

Query processing is accomplished by repeated computation and updates of the sensor dependency tree. During each iteration, one or more nodes are selected for computation. The selected nodes must *not* be dependent on any other nodes. After the computation, one or more nodes are removed from the sensor dependency tree. The process then iterates and eventually the last node in the tree is reached; the last node of the dependency tree is generally the fusion node, which performs the fusion operation. After the fusion operation is carried out the process terminates. Fusion in ΣQL is based on a voting approach [9]. The motivation for this fusion approach is that it is fast which is necessary requirement in a query language. However, the fusion method is interchangeable. A further aspect is that the fusion method must be well integrated with the query system such that the end-users do not have to bother with how the fusion method behaves. To accomplish this, there must be methods available that will support the end-users in judging the result of a fused query; this is discussed further in connection to the confidence values in section 5.

3.3 Multi-Level View Database

A multi-level view database (MLVD) is needed to support sensor-based query processing. The status information is obtained from the sensors, which includes the object type, the attribute values such as colour or length, status information of type position, orientation, and so on. The positions of the query originator and the sensors may also change. This information is processed and integrated into the multi-level view database. Whenever the query processor needs some information, it asks the view manager which is the subsystem that maintains the view database. The view manager also shields the rest of the system from the details of managing sensor data, thus achieving sensor data independence.

The multiple views may include the following three views in a resolution pyramid like structure: the global view, the local view and the object view. The *global view* describes where the target object is situated in relation to some other objects, e.g. a road from a map. This will enable the sensor analysis program to find the location of the target object with greater accuracy and thus make a better analysis. The *local view* provides the information such as whether the target object is partially hidden. The local view can be described, for example, in terms of Symbolic Projection [10]. Finally, there is also a need for a *symbolic object description,* which describes the target itself in great detail. The three views may include information about the query originator and can be used later on in other tasks such as in situation analysis [10].

The view manager can be regarded as an agent, or as middleware, depending upon the system architecture. The multi-level views are managed by the view manager. The global view is obtained primarily from a geographic information system (GIS). The local view and the object view are more detailed descriptions of local areas and objects. The results of query processing, and the movements of the query originator, may both lead to the updating of all three views.

3.4 Query Refinement

There is another class of queries that require more sophisticated query refinement. We will call this class of queries *evolutionary queries*. An evolutionary query is a query that may change over time and/or in space. For example when an emergency management worker moves around in a disaster area, a predefined query can be executed repeatedly to evaluate the surrounding area to find objects of threat. However, queries and query processing are affected by the spatial/ temporal relations among the query originator, the sensors and the sensed objects.

Given a user query in a high-level language, i.e. either the natural language or a visual language forms the query refinement approach that is outlined below, where *italic words* indicate operations for the second (and subsequent) iteration.

Step 1. Analyze the user query to generate/*update* the sensor dependency tree based upon the ontological knowledge base and the multi-level view database that contains up-to-date contextual information in the object view, the local and the global views.

Step 2. If the sensor dependency tree is empty then terminate otherwise if only the fusion node remains, perform the fusion operation and terminate. Otherwise build/*refine* the σ-query, i.e. the internal query, based upon the user query, the sensor dependency tree and the multi-level view database.

Step 3. Execute the portion of the σ-query that is executable according to the sensor dependency tree.

Step 4. Update the multi-level view database and go back to Step 1.

In query processing/refinement, the spatial/temporal relations must be taken into consideration in the construction/update of the sensor dependency tree, which is controlled by the ontology system. The temporal relations such as “followed by”, “preceded by” should be allowed while the spatial relations should include the common spatial relations, see e.g. [11]. Other special relations such as “occluded by”, and so on must be available as well.

4.0 THE ONTOLOGY SYSTEM

The purpose of the sensor data independence concept introduced above is to simplify the use of the system and to let the system take the responsibility of deciding which sensors and which sensor data analysis algorithms that should be applied under given circumstances in response to a particular query. To support this activity an ontological knowledge base system (OKBS) has been developed. This is a step towards a general technique to generate/refine queries based upon incomplete knowledge about the real world. However, the knowledge stored in the ontology differs from knowledge in other domains in that it includes not just object knowledge but sensor and sensor data control knowledge as well.

The ontology is taxonomically divided into three parts: the *sensor & algorithm part* describing the sensors and recognition algorithms, the *conditions part* describing external conditions such as weather and light conditions and the *thing-to-be-sensed part* describing the objects and the object properties to be sensed. In figure 1 a simplified overview of the ontology is presented. The concepts in the *sensor & algorithm part* are presented to the left of the *ThingToBeSensed* concept. The *conditions part* is to the right of the *ThingToBeSensed* concept and the *ThingToBeSensed* concept itself together with its subconcepts make up the *thing-to-be-sensed part*.

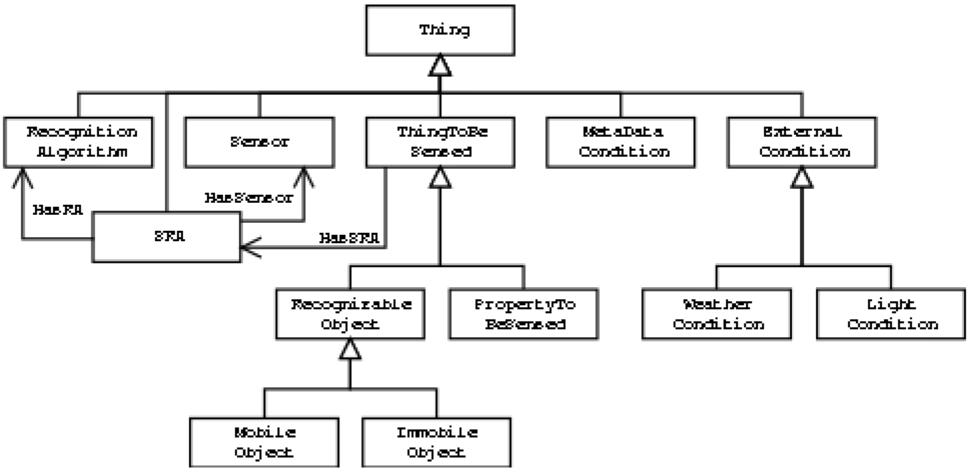


Figure 1: Ontology Overview Describing the General Knowledge Structure.

Using this knowledge in conjunction with information about the objects to be sensed and rules describing under what conditions certain sensors and recognition algorithms are appropriate to use, the OKBS determines which sensors and recognition algorithms to use under the given conditions. For example, IR (infrared) and LR (laser radar) sensors can be used at night, while CCD (digital camera) cannot. Probably, IR can be used in a foggy weather, but LR and CCD cannot, etc.

Most of the concepts in the simplified ontology overview in figure 1 are self-explanatory. The *SRA* concept models a many-to-many relation between sensors and recognition algorithms, meaning that a certain recognition algorithm can be used on data from different kinds of sensors and many different recognition algorithms can be used on data from a certain sensor type. This means that each SRA can be used to determine a combination of a sensor and a recognition algorithm that work well together. The *HasRA* and *HasSensor* relations are used to model this.

MetaData condition is a concept that models meta-data conditions like data quality. *PropertyToBeSensed* models properties that the sensors can sense, e.g. color, temperature, etc. *RecognizableObject* models all objects that the recognition algorithms can recognize.

Relations and concept attributes are inherited down the inheritance chain, meaning that the *HasSRA* relation applies not only to the *ThingToBeSensed* concept. It also applies to all its subconcepts, e.g. *RecognizableObject*, *MobileObject*, etc.

The *HasSRA* relation describes which combinations of sensors and recognition algorithms are the most appropriate to use under ideal external conditions (weather, time of day, etc). When the OKBS generates/refines a query it first uses this relation to find out which sensors and recognition algorithms that are appropriate when trying to recognize the requested object type(s) under ideal conditions.

The next step is to take meta-data conditions such as sensor data quality into account. The sensors and recognition algorithms selected in the first step are re-evaluated with respect to their respective meta-data quality at the time and place indicated in the query.

Finally, the external conditions are taken into account. To perform this step a rule based system has been developed. Given the external conditions at the time and place indicated in the query, this step re-evaluates the selected sensors and recognition algorithms according to the rules that describe how appropriate the selected sensors and recognition algorithms are under certain conditions. Each rule describes the appropriateness of a certain sensor or recognition algorithm given a complete set of external conditions.

The result of the described process is a prioritized list of appropriate combinations of sensors and recognition algorithms to use for the given query under the external conditions at the time and place of the query. This information is used to construct/refine the sensor dependency tree that in turn determines in which order different parts of the query should be processed.

When the number of sensor types and recognition algorithms grow the number of rules will also grow. At present, the rules are updated manually, but as more sensor data becomes available from different test scenarios it will be possible to develop a system that can tune the rules in a semi-automatic manner by means of mathematical statistics and artificial intelligence techniques.

The ontological knowledge base system is described in much more detail in [12] where complete descriptions of all the concepts in the ontology can be found.

5.0 THE USER INTERFACE

In current sensor-based information systems detailed knowledge about the sensors is required. Therefore sensor selection is left to the users who supposedly are also experts on sensors. However, in real life this is not always the case. A user cannot be an expert on all sensors and all sensor data types. Therefore query systems with the ability to hide this kind of low-level information from the users need to be developed. User interfaces also need to be designed to allow the users to formulate queries with ease and request information at a high-level of abstraction to accomplish sensor data independence. An approach to overcome these problems and to accomplish sensor data independence is proposed through the use of the sensor dependency tree, the query refinement technique, the multi-level view databases, and an ontological knowledge base for the sensors and objects to be sensed.

One of the advantages with this system is that the user is not restricted to querying the system about things that one single sensor may answer. He can also make queries that use the combined information from sensors. He can for instance ask for the location of all blue vehicles that have their engines running. This query requires information both from an infrared camera to see if the engine is hot and from a camera in the visual range to see if the vehicle is blue. The user does not have to feel restricted to the information a single sensor might be able to provide, but is free to make the queries that he needs an answer of and the system will find out which sensors that are appropriate to use.

To run a query of general type the user needs an interface where he can select which area he is interested in (AOI), the relevant period in time (IOI) and what kind of objects he is looking for. Other query types are allowed as well and could be made up by combinations of these three concepts including required property conditions. However, the user will not be given a choice of sensors, instead information will be available on which areas and which times are covered by any of the sensors. The user will have the possibility to visually specify attributes of objects and relations between the objects to be able to make more advanced queries.

The ontological knowledge base provides the user interface with all objects and attributes that can be queried i.e. all attributes that at least one sensor can recognize. That way, the user has access to all currently available options without having to know anything about the available sensors and algorithms.

As described in previous sections, some queries such as the evolutionary queries need to be processed repeatedly, with minor changes, during a certain period of time (time interval of interest) and/or within a certain geographical area (area of interest). Since most users of sensor-based information systems are not experts in sensors, we propose to attack the problem in two ways. By constructing an ontological knowledge base, where the low-level detailed information about sensors, objects to be sensed and environmental conditions can be stored. By providing query templates, the commonly encountered queries can be specified by e.g. form-filling. To formulate such queries a template can be used accordingly:

Step 1. The user enters the selected dimensions, the ordering of the projections, the sources, and the join conditions.

Step 2. Dimensions, sources, join conditions can all be based upon selections from pull-down menus.

Step 3. A query template is filled in to generate the query.

Step 4. In the WHERE part, if the type of an object is set to 'aaa', there must be an algorithm to recognize the object 'aaa', or the object 'aaa' is already in the database. This can be determined by checking with the ontological knowledge.

Step 5. The query processor processes the σ-join-query.

To support evolutionary queries, a further refinement of the technique is to let the user specify queries with additional parameters, such as how often to run the query, in what time intervals and geographical areas, and so on. Since evolutionary queries usually change slowly, we will investigate techniques to optimize evolutionary query processing, by maximizing the sharing of information in the processing of consecutive evolutionary queries.

A general and most important aspect of any query system and particularly in sensor data fusion systems, is the confidence in the query result, which must be acknowledged by the user. This is due to the fact that data acquired from sensors are always mapping the reality with some level of uncertainty. The uncertainties are due to, among other things, technical imperfections in the sensors. Generally, these uncertainties can be represented with some kind of confidence value that may be normalized, i.e. within the interval [0,1]. Confidence values should be interpreted as the confidence a user may have in a query result. This way of representing uncertainties in the data becomes even more necessary in the sensor data fusion processes. Consequently, when evaluating the result from a query applied to data from multiple sensors the confidence values corresponding to the uncertainties of the fused result is required. This kind of confidence structure is used in ΣQL to support the user in interpreting the query result.

The result of the query is presented on a map that covers the given AOI. All objects found are presented by using the standardized unit symbols. At the user's request all available information about an object will be displayed, i.e. color, number of antennas and the deduction behind the sensor information fusion. To get an intuitive feeling of how reliable the result is the symbols will be color coded with respect to the confidence values, i.e. strong color – high confidence, weak color – low confidence values.

The map view is a good way to get a spatial overview of the result. It is also similar to what is done in classical C_2-systems. In some cases spatial overview is not the optimal way, so in addition to the map view the user has the option to get the information in a spreadsheet oriented way where the user has the option to sort the data in any way he wants.

6.0 CONCLUSIONS

In this paper, a query language for heterogeneous data sources, generally corresponding to various types of sensors, has been introduced. An important characteristic of this query language is the concept of sensor data independence. In particular, there are three important aspects of sensor data independence which all have a strong influence on the users' working situation. Thus the consequences of this concept are:

- The users do not need to have any knowledge about the sensors that are used to answer a particular query.
- For a query that repeatedly is applied over a fairly long period in time sensors can be engaged/ disengaged without the users knowledge depending on e.g. weather or light conditions.

- New sensor types and recognition algorithms can be added/deleted to the system, when ever suitable, without informing the users.

Most of these aspects depend on the ontological knowledge-base system and furthermore, this subsystem also has an influence on the query refinement concept that is supported by the introduction of the global, local and object views.

A further characteristic, that is of vital interest to the users, is that the users by the introduction of the sensor data independence concept are able to use application dependent notions in their queries, e.g. area of interest, object type, relations etc. These notions will make it possible to design a powerful visual user interface or language suitable to the ΣQL query language.

An important aspect that concerns the sensor data fusion part of the system is that fusion is an integrated part of the query language that is of less concern to the end-users. Thus, from the users perspective, of concern is how to handle the confidence values that are given as complementary parts of the query results. A question of concern in relation to these confidence values is how to present this information to the end-users in a way that is simple to interpret. For example, do we believe in the given query result or not.

Future research of concern in this work will heavily be focusing on the design of the visual user interface and further development of the OKBS in order to determine improvements of the concept of sensor data independence.

7.0 REFERENCES

[1] E. Waltz and J. Llinas, *Multisensor Data Fusion*, Artect House, Boston, 1990.

[2] *Handbook of Multisensor Data Fusion*, D.L. Hall, J. Llinas (Eds.), CRC Press, London, 2001.

[3] S.-K. Chang, G. Costagliola, E. Jungert, *Spatial/Temporal Query Processing for Information Fusion Applications*, Advances in Visual Information Systems, R. Laurini (Ed.), Lecture Notes in Computer Science 1929, Springer Verlag, Berlin, 2000, pp. 127-139.

[4] S.-K. Chang, G. Costagliola, E. Jungert, *Multisensor Information Fusion by Query Refinement*, Recent advances in Visual information systems, S.-K. Chang, Z. Chen, S.-Y. Lee (Eds.), Lecture Notes in Computer Science 2314, Springer Verlag, Berlin, 2002, pp. 1-11.

[5] J.D. Ullman, "Database and Knowledge-Base Systems", Vol. 1, Computer Science Press, Rockville, Maryland, USA, 1988, pp. 11-12.

[6] C. Date, *An Introduction to Database Systems*, Addison-Wesley, 1995.

[7] http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[8] M. Jarke and J. Cohen, "Query Optimization in Database Systems", ACM Computing Surveys, Vol. 16, No. 2, 1984.

[9] J.R. Parker, "*Multiple Sensors, Voting Methods and Target Value Analysis*", Proceedings of SPIE Conference on Signal Processing, Sensor Fusion and Target Recognition VI, SPIE Vol. 3720, Orlando, Florida, April 1999, pp. 330-335.

[10] S.-K. Chang and E. Jungert, *Symbolic Projection for Image Information Retrieval and Spatial Reasoning*, Academic Press, London, 1996.

[11] S.Y. Lee and F.J. Hsu, "Spatial Reasoning and Similarity Retrieval of Images using 2D C-string Knowledge Representation", Pattern Recognition, Vol. 25, No 3, 1992, pp. 305-318.

[12] T. Horney, "Design of an Ontological Knowledge Structure for a Query Language for Multiple Data Sources", FOI-R--0498--SE, Linköping, Sweden, 2002.

SYMPOSIA DISCUSSION – PAPER NO: 10

Author Name(s):

Dr. Erland Jungert, FOI, Linkøping, Sweden
Karin Silvervarg, FOI, Linkøping, Sweden

Question:
Targets are color coded with respect to confidence values, but there are different types of uncertainty such as where or what type. What is the color referring to in this case?

Author's Response:
In this case, the uncertainty has to do with whether or not there is a target in the location indicated.

Question:
Collection management is currently a manual process that takes a long time to task a series of sensors to obtain data. Have you given any thoughts on a process to automate the process?

Author's Response:
In a real situation, it has to be trained. Also looking at selecting the sensor that has the best value available.

Ontology driven sensor independence in a query supported C_2-system

Erland Jungert,
Karin Silvervarg,
Tobias Horney
Swedish Defence Research Agency (FOI)

FOI

Background

• Information systems requiring sensor data input must generally include means for sensor data fusion as well as powerful mechanisms for user interaction and result visualization.

• From a user perspective the use of sensor data requires knowledge about the attached sensors and the data they generate.

• Most end-users do not possess this kind of knowledge and for this reason techniques for sensor data independence must be developed to ensure that queries can be applied and the results interpreted in a convenient way.

Sensor data independence

- Sensor data independence (SDI) relates to data independence as introduced in database design for modification of the physical databases without affecting the application programs.
- SDI is required from a user perspective to simplify the use of large data quantities generated by multi-sensor data systems.
- SDI diminishes the workload of the users and the queries becomes simpler to apply.
- A consequence of the concept is that the users must learn to trust the system and the returned results.

Sensor data independence

• New sensor types can be integrated without informing the users by updating the ontological knowledge-base.

•The approach taken to establish SDI is through the introduction of an *ontology* which here is used to support sensor oriented queries in a query language called ΣQL.

•Information of concern to the ontology is *requested objects, the actual weather conditions, the time of the day* and *the available sensor data*

• The users do not have to know which sensors that are used to answer a query. Consequently, the sensor types may vary over time in an optimized way.

FOI

Characteristics of ΣQL

- Input to ΣQL are heterogeneous data from sensors.
- The sensor data are transformed to a unified spatial/temporal information structure which can be fused when needed.
- The internal information structure can also be used for spatial reasoning, e.g. in situation analysis.
- Queries are made up through a sequence of spatial /temporal operators, called *σ-operators*.
- Queries are applied in terms of *AOI, IOI, targets, target attributes* and *target relations.*

FOI

Query types

The QL should be able to answer queries of the type:

Has there been any activity in the AOI during the last 2 hours

Show all targets in the AOI between now and midnight; respond every 15 minutes.

An example including data fusion

$\phi_{xyt}(*)$

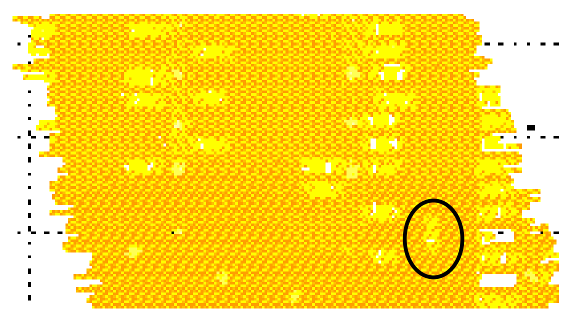
$(\sigma_{motion}(moving)\sigma_{type}(vehicle)\ \sigma_{xy,interval\_cutting}(*)$

$\sigma_t(T^o)_{T \bmod 10 = 0 \text{ and } T>t1 \text{ and } T<t2}$

$\sigma_{media\_sources}\ (video^o)media\_sources,$

$\sigma_{type}\ (vehicle)\ \sigma_{xyz,interval\_cutting}(*)\sigma_t(T^o)_{\ T>t1 \text{ and } T<t2}$

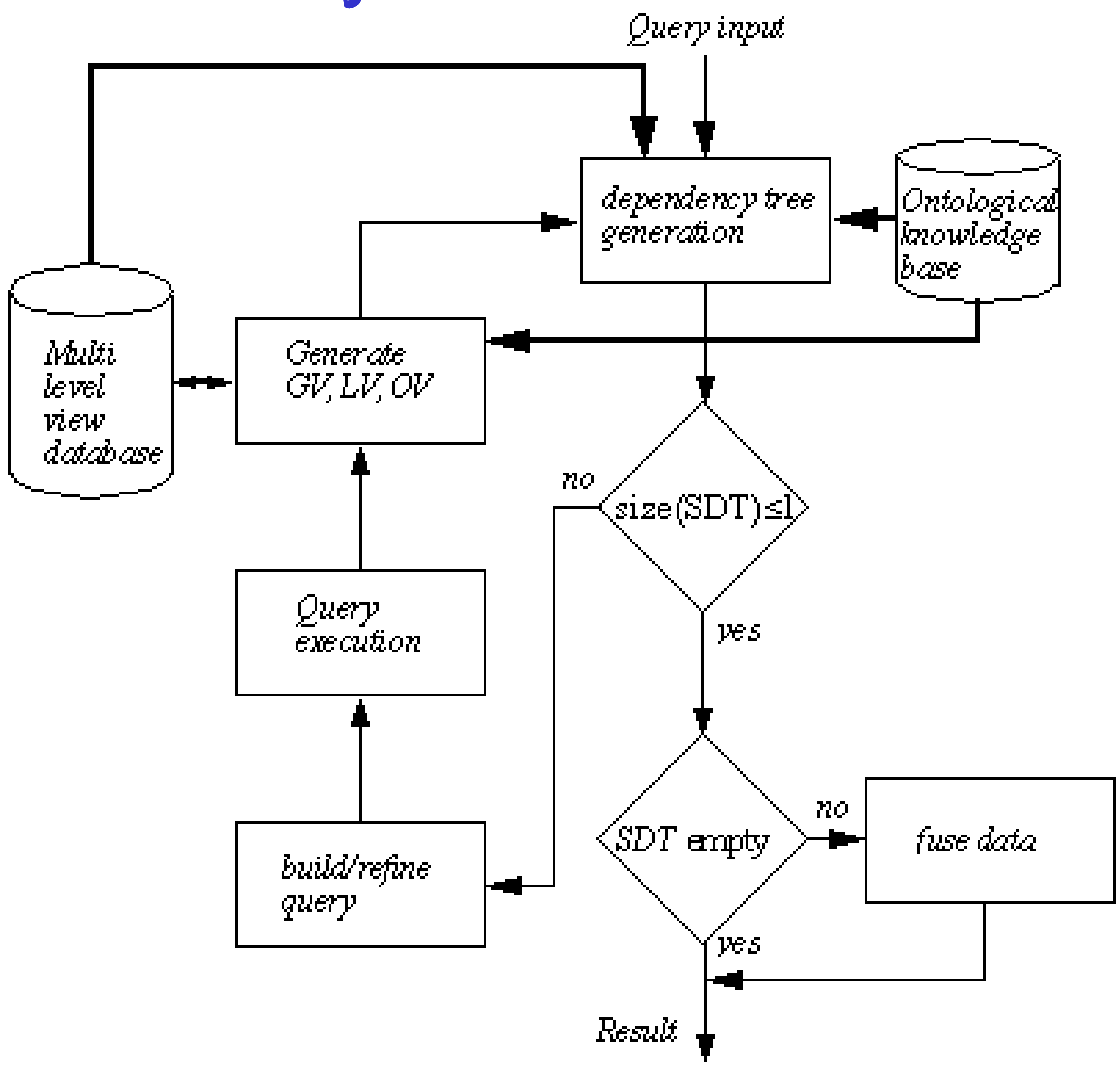
$\sigma_{media\_sources}(laser\_radar^{o)}\ media\_sources)$



Special σ–operators

- Object classification
- Attribute determination
- Determination of status values, such as
  - object positions
  - tracks
  - orientation
  - speed
- Object relations, e.g.
  - direction
  - distance

The system structure



FOI

The ontological knowledge-base

The purpose of the ontology is

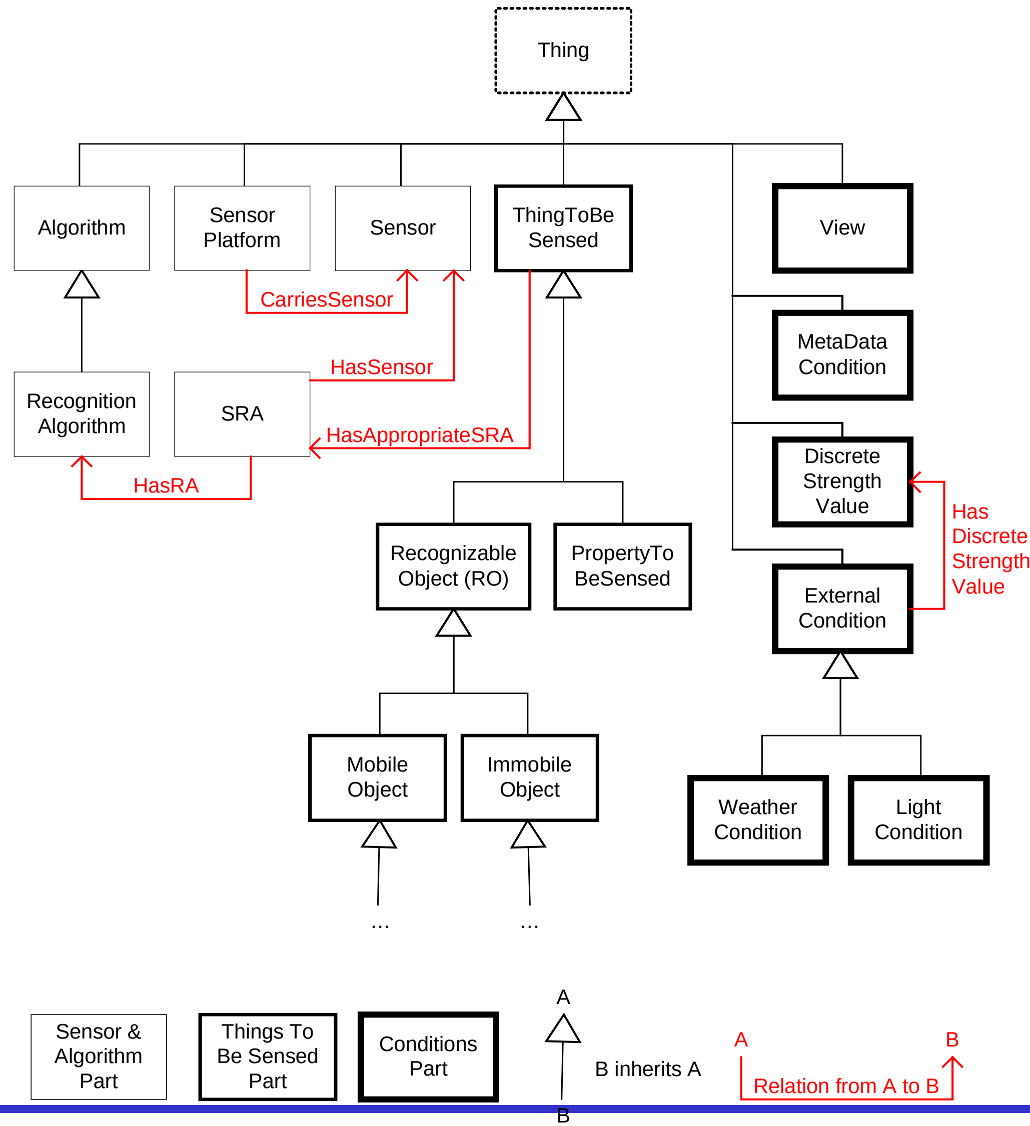
- to achieve sensor data independence,
- to support query refinement.

Things to consider with respect to the ontology:

- What is asked for in the query.
- The area of interest (AOI).
- Sensor and algorithm characteristics.
- Meta data including data quality.
- Weather and light conditions.
- Terrain background.

FOI

The structure of the ontology



Algorithm/sensor determination

- A special algorithm has been developed to support the determination of the most appropriate sensors and recognition algorithms in the given query.
- From the ontology suitable SRAs and initial appropriate values are determined for the actual object type(s).
- All factors of importance, e.g. weather etc, are weighted together and a final value of appropriateness is determined for each SRA.

Rules

• Describes which sensors and recognition algorithms that are suitable and under which conditions.

If *factor* ***X*** *has the value* ***Y then*** *its influence on sensor/algorithm* ***Z*** *has the value* ***V***

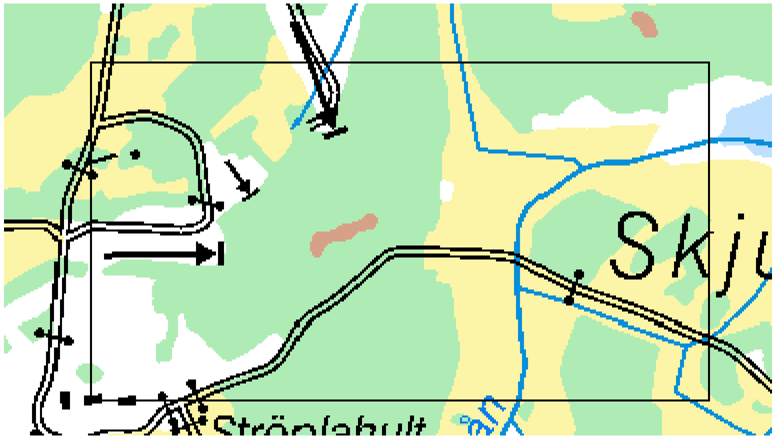
If *factor* ***rain*** *has the value* ***gentle then*** *its influence on sensor/algorithm* ***vehicle-alg*** *has the value* ***little (0.2)***

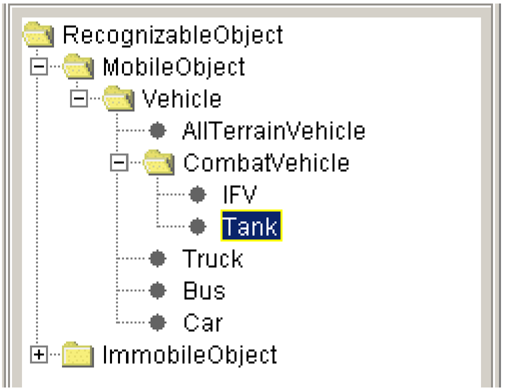
FOI

Querying the system

The concepts the user uses are:

- Area of interest
- Interval of interest
- Object types
- Property conditions





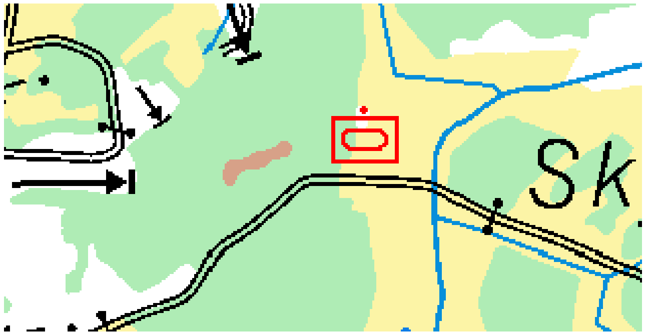
FOI

Confidence values

- There is always uncertainty
- The system calculates a confidence value
- Normalized to be in the interval [0,1]
- Describes the confidence a user may have in a query result

Result presentation

- Presented on a map
- Visualised with standardized unit symbols
- The user can request all available information about an object
- Targets are colour-coded with respect to confidence values
- Possibility of getting the result in a spreadsheet oriented way



FOI

Conclusions

• The users do not need to have any knowledge about the sensors that are used in a particular query.

• For a query that repeatedly is applied over a fairly long period in time sensors can be engaged/ disengaged without the users' knowledge depending on e.g. weather or light conditions.

• New sensor types and recognition algorithms can be added/deleted when ever suitable; without informing the users.